

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN**



**BÁO CÁO BÀI THỰC HÀNH
HỌC PHẦN: THỰC TẬP CƠ SỞ
MÃ HỌC PHẦN: INT13147**

**BÀI THỰC HÀNH 2.2
TÌM HIỂU VÀ CÀI ĐẶT, CẤU HÌNH NIDS**

Sinh viên thực hiện:

B22DCAT253 Đinh Thị Thanh Tâm

Giảng viên hướng dẫn: TS. Đinh Trường Duy

HỌC KỲ 2 NĂM HỌC 2024-2025

MỤC LỤC

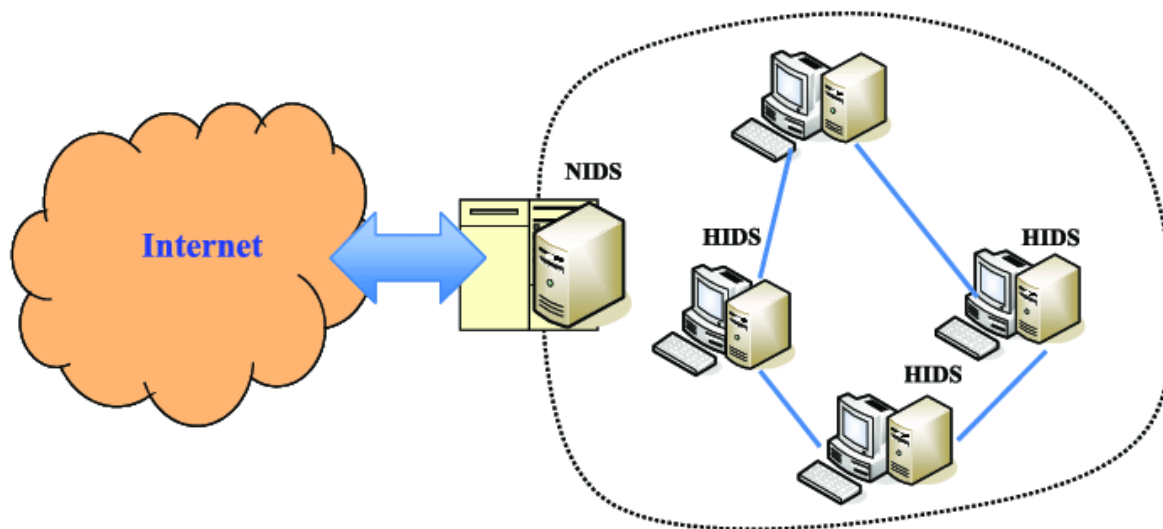
MỤC LỤC.....	2
CHƯƠNG 1. LÝ THUYẾT BÀI THỰC HÀNH.....	3
1.1 Mục đích.....	3
1.1 Tìm hiểu khái quát về các hệ thống phát hiện tấn công, xâm nhập	3
1.2 Tìm hiểu về kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập, như Snort, Suricata, Zeek, OSSEC, Wazuh... ..	4
1.2.1 Snort	4
1.2.2 Suricata.....	5
1.2.3 Zeek (Bro IDS).....	5
1.2.4 OSSEC	5
1.2.5 Wazuh.....	6
1.3 Snort (sơ đồ khối, các thành phần, luật).....	6
1.3.1 Giới thiệu về Snort?	6
1.3.2 Kiến trúc của Snort.....	7
1.3.3 Bộ luật của Snort	7
CHƯƠNG 2. NỘI DUNG THỰC HÀNH	10
2.1 Chuẩn bị môi trường	10
2.2 Các bước thực hiện.....	10
Bước 1: Chuẩn bị hệ thống.....	10
Bước 2: Cài đặt và chạy thử Snort	11
Bước 3: Viết và triển khai các quy tắc phát hiện tấn công.....	11
Bước 4: Thực hiện tấn công kiểm thử.....	12
KẾT LUẬN	17
TÀI LIỆU THAM KHẢO.....	17

CHƯƠNG 1. LÝ THUYẾT BÀI THỰC HÀNH

1.1 Mục đích

- Tìm hiểu và luyện tập việc cài đặt và vận hành các hệ thống phát hiện xâm nhập cho host (HIDS) và cho mạng (NIDS).
- Luyện tập việc tạo và chỉnh sửa các luật phát hiện tấn công, xâm nhập cho các hệ thống phát hiện xâm nhập thông dụng.

1.1 Tìm hiểu khái quát về các hệ thống phát hiện tấn công, xâm nhập



NIDS: Network Intrusion Detection Systems thường được bố trí tại những điểm dễ bị tấn công trong hệ thống. NIDS được sử dụng để giám sát traffic đến và đi từ tất cả các thiết bị trên mạng. Điểm cộng lớn nhất của NIDS là có thể quét tất cả traffic inbound và outbound, nhưng việc này có thể làm giảm tốc độ chung của mạng.

HIDS: Host Intrusion Detection Systems, hệ thống phát hiện xâm nhập này hoạt động trên tất cả các thiết bị trong hệ thống có thể kết nối Internet. HIDS chỉ giám sát các gói dữ liệu inbound và outbound từ thiết bị hoặc những hành động đáng ngờ tại cấp truy cập nội bộ.

Signature-Based: Đây là các IDS hoạt động dựa trên chữ ký, giám sát các gói tin trên mạng tương tự như cách phần mềm diệt virus hoạt động. Tuy nhiên Signature-Based có thể không phát hiện được những mối đe dọa mới, khi chữ ký để nhận biết nó chưa được IDS cập nhật.

Anomaly-Based: IDS này được sử dụng để phát hiện mối đe dọa dựa trên sự bất thường. Anomaly-Based sẽ giám sát traffic mạng và so sánh với baseline đã được thiết lập từ trước. Baseline sẽ xác định đâu là mức bình thường của mạng và cảnh báo cho quản trị viên mạng hoặc người dùng khi phát hiện traffic truy cập bất thường hoặc khác biệt so với baseline.

Passive: Đây là IDS thụ động chỉ phát hiện và cảnh báo. Khi phát hiện traffic đáng ngờ hoặc độc hại, nó sẽ tạo và gửi cảnh báo đến các nhà quản trị hoặc người dùng. Những hành động sau đó sẽ phụ thuộc vào người quản trị.

Reactive: Loại IDS này ngoài nhiệm vụ như IDS Passive, nó còn thực hiện những hành động đã được thiết lập sẵn để phản ứng lại các mối đe dọa một cách nhanh chóng, ví như: chặn nguồn truy cập, khóa IP.

Ưu điểm của IDS:

- Thích hợp sử dụng để thu thập số liệu, giúp kiểm tra các sự cố xảy ra đối với hệ thống mạng với những bằng chứng thuyết phục nhất.
- Đem đến cái nhìn bao quát và toàn diện về toàn bộ hệ thống mạng.
- Là công cụ thích hợp để thu thập bằng chứng phục vụ cho việc kiểm tra các sự cố trong hệ thống mạng.

Nhược điểm của IDS:

- Nếu không được cấu hình hợp lý rất dễ gây tình trạng báo động nhầm.
- Khả năng phân tích lưu lượng mã hóa tương đối thấp.
- Chi phí triển khai, phát triển và vận hành hệ thống tương đối lớn.

1.2 Tìm hiểu về kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập, như Snort, Suricata, Zeek, OSSEC, Wazuh...

1.2.1 Snort

1.2.1.1 Kiến trúc

Snort là một hệ thống phát hiện và ngăn chặn xâm nhập (IDS/IPS) dựa trên quy tắc, hoạt động theo mô hình kiểm tra gói tin mạng.

Gồm 4 thành phần chính:

- Bộ thu thập gói tin: Thu thập dữ liệu mạng từ giao diện mạng.
- Bộ tiền xử lý (Preprocessor): Phân tích, chuẩn hóa và tối ưu dữ liệu trước khi kiểm tra.
- Bộ kiểm tra quy tắc (Detection Engine): So sánh các gói tin với tập luật phát hiện tấn công.
- Bộ ghi log và cảnh báo: Lưu lại thông tin khi phát hiện mối đe dọa.

1.2.1.2 Tính năng

- Hỗ trợ phát hiện tấn công theo chữ ký (signature-based) và theo hành vi bất thường.
- Có khả năng phân tích và lọc gói tin theo nhiều giao thức mạng.
- Hỗ trợ ghi log chi tiết và có thể tích hợp với các công cụ SIEM.
- Cho phép chặn tấn công bằng chế độ IPS khi kết hợp với tường lửa.

1.2.2 Suricata

1.2.2.1 Kiến trúc

- Suricata là một IDS/IPS hiện đại, có khả năng phân tích lưu lượng mạng sâu hơn Snort.
- Cấu trúc tương tự Snort nhưng cải tiến hơn với khả năng xử lý song song và hiệu suất cao.
- Sử dụng công nghệ Multi-Threading, giúp tận dụng tối đa tài nguyên CPU.

1.2.2.2 Tính năng

- Hỗ trợ phát hiện tấn công bằng quy tắc, phân tích hành vi, phát hiện dựa trên bất thường (anomaly-based detection).
- Hỗ trợ kiểm tra lưu lượng TLS, HTTP, DNS mà không cần giải mã.
- Tích hợp mạnh mẽ với các hệ thống SIEM, có thể ghi log theo chuẩn JSON.
- Hỗ trợ phát hiện mã độc bằng cách quét payload của các gói tin.

1.2.3 Zeek (Bro IDS)

1.2.3.1 Kiến trúc:

Zeek hoạt động như một Network Security Monitor (NSM), phân tích lưu lượng mạng theo phiên (session-based) thay vì kiểm tra từng gói tin.

Bao gồm:

- Core Analysis Framework: Phân tích dữ liệu mạng và lưu trữ kết quả.
- Policy Scripts: Định nghĩa quy tắc phát hiện xâm nhập bằng ngôn ngữ scripting của Zeek.

1.2.3.2 Tính năng:

- Phân tích sâu các giao thức như HTTP, FTP, DNS, SSL/TLS.
- Có khả năng phát hiện hành vi bất thường bằng cách thu thập và phân tích metadata của lưu lượng mạng.
- Cung cấp thông tin phong phú hơn IDS truyền thống nhờ vào việc ghi log chi tiết.
- Có thể kết hợp với Suricata hoặc Snort để tăng cường khả năng phát hiện tấn công.

1.2.4 OSSEC

1.2.4.1 Kiến trúc:

OSSEC là một HIDS (Host-based Intrusion Detection System), hoạt động bằng cách giám sát tệp nhật ký hệ thống, kiểm tra tính toàn vẹn của tệp, và phân tích hành vi trên máy chủ.

Gồm 3 thành phần chính:

- OSSEC Agent: Cài đặt trên các máy chủ, thu thập dữ liệu và gửi về máy chủ trung tâm.
- OSSEC Manager: Phân tích dữ liệu, kiểm tra quy tắc và tạo cảnh báo.
- OSSEC Web UI: Giao diện hiển thị kết quả giám sát và quản lý tập trung.

1.2.4.2 Tính năng:

- Giám sát file hệ thống để phát hiện thay đổi bất thường.
- Phân tích log hệ thống để tìm dấu hiệu tấn công.
- Hỗ trợ phát hiện rootkit, kiểm tra registry trên Windows.
- Có thể tích hợp với SIEM như Splunk hoặc ELK Stack để phân tích dữ liệu.

1.2.5 Wazuh

1.2.5.1 Kiến trúc:

Wazuh là phiên bản mở rộng của OSSEC, kết hợp với ELK Stack để thu thập, phân tích và trực quan hóa dữ liệu bảo mật.

Thành phần chính:

- Wazuh Agent: Thu thập dữ liệu từ máy chủ.
- Wazuh Server: Kiểm tra và xử lý thông tin bảo mật.
- Elastic Stack: Hiển thị và quản lý dữ liệu bảo mật trên giao diện web.

1.2.5.2 Tính năng:

- Cung cấp đầy đủ các chức năng của OSSEC với khả năng mở rộng cao hơn.
- Hỗ trợ kiểm tra bảo mật dựa trên tiêu chuẩn như PCI DSS, GDPR.
- Có thể phát hiện mã độc, rootkit, tấn công brute force.
- Cho phép giám sát container và môi trường cloud như AWS, Azure, GCP.

1.2.5.3 So sánh nhanh các hệ thống IDS

Hệ thống	Loại IDS	Đặc điểm chính
Snort	NIDS	IDS/IPS dựa trên quy tắc, dễ triển khai
Suricata	NIDS	Xử lý đa luồng, hiệu suất cao, phát hiện mã độc
Zeek	NIDS	Phân tích theo phiên, log chi tiết
OSSEC	HIDS	Giám sát file, log, rootkit
Wazuh	HIDS	OSSEC + ELK Stack, hỗ trợ cloud

1.3 Snort (sơ đồ khối, các thành phần, luật)

1.3.1 Giới thiệu về Snort?

Snort là một NIDS được Martin Roesh phát triển dưới mô hình mã nguồn mở. Tuy Snort miễn phí nhưng nó lại có rất nhiều tính năng tuyệt vời mà không phải sản phẩm thương mại nào cũng có thể có được. Với kiến trúc thiết kế theo kiểu module, người dùng có thể tự tăng cường tính năng cho hệ thống Snort của mình bằng việc cài đặt hay viết thêm mới các module. Cơ sở dữ liệu luật của Snort đã lên tới 2930 luật và được cập nhật thường xuyên bởi một cộng đồng người sử dụng. Snort có thể chạy trên nhiều hệ thống nền như Windows, Linux, OpenBSD, FreeBSD, NetBSD, Solaris, HP-UX, AIX, IRIX, MacOS.

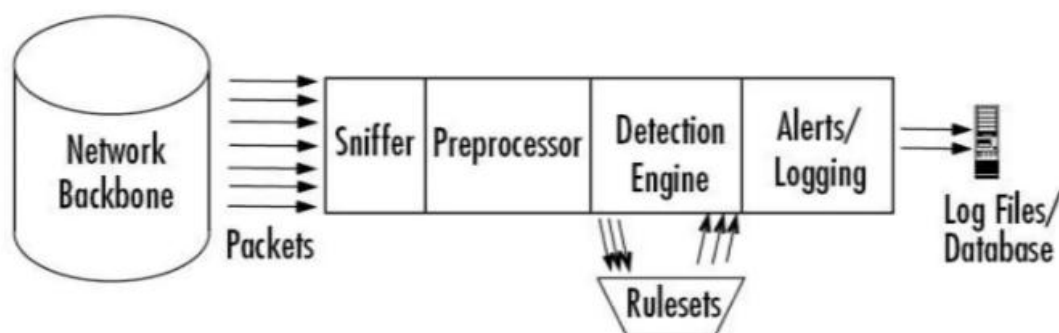
Bên cạnh việc có thể hoạt động như một ứng dụng thu bắt gói tin thông thường, Snort còn có thể được cấu hình để chạy như một NIDS. Snort hỗ trợ khả năng hoạt động trên các giao thức sau: Ethernet, 802.11, Token Ring, FDDI, Cisco HDLC, SLIP, PPP, và PF của OpenBSD.

1.3.2 Kiến trúc của Snort

Snort bao gồm nhiều thành phần, với mỗi phần có một chức năng riêng. Các phần chính đó là:

- Module giải mã gói tin (Packet Decoder)
- Module tiền xử lý (Preprocessors)
- Module phát hiện (Detection Engine)
- Module log và cảnh báo (Logging and Alerting System)
- Module kết xuất thông tin (Output Module)

Kiến trúc của Snort được mô tả trong hình sau:



Khi Snort hoạt động nó sẽ thực hiện việc lắng nghe và thu bắt tất cả các gói tin nào di chuyển qua nó. Các gói tin sau khi bị bắt được đưa vào Module Giải mã gói tin. Tiếp theo gói tin sẽ được đưa vào Module Tiền xử lý, rồi Module Phát hiện. Tại đây tùy theo việc có phát hiện được xâm nhập hay không mà gói tin có thể được bỏ qua để lưu thông tiếp hoặc được đưa vào Module Log và cảnh báo để xử lý. Khi các cảnh báo được xác định Module Kết xuất thông tin sẽ thực hiện việc đưa cảnh báo ra theo đúng định dạng mong muốn.

1.3.3 Bộ luật của Snort

1.3.3.1 Giới thiệu

Cũng giống như virus, hầu hết các hoạt động tấn công hay xâm nhập đều có các dấu hiệu riêng. Các thông tin về các dấu hiệu này sẽ được sử dụng để tạo nên các luật cho Snort. Thông thường, các bẫy (honey pots) được tạo ra để tìm hiểu xem các kẻ tấn công làm gì cũng như các thông tin về công cụ và công nghệ chúng sử dụng. Và ngược lại, cũng có các cơ sở dữ liệu về các lỗ hổng bảo mật mà những kẻ tấn công muốn khai thác.

Các dạng tấn công đã biết này được dùng như các dấu hiệu để phát hiện tấn công xâm nhập. Các dấu hiệu đó có thể xuất hiện trong phần header của các gói tin hoặc nằm trong phần nội dung của chúng. Hệ thống phát hiện của Snort hoạt động dựa trên các luật (rules) và các luật này lại được dựa trên các dấu hiệu nhận dạng tấn công. Các luật có thể được áp dụng cho tất cả các phần khác nhau của một gói tin dữ liệu.

Một luật có thể được sử dụng để tạo nên một thông điệp cảnh báo, log một thông điệp hay có thể bỏ qua một gói tin.

1.3.3.2 Cấu trúc luật của Snort

Hãy xem xét một ví dụ đơn giản:

```
alert tcp 192.168.2.0/24 23 -> any any (content:"confidential"; msg: "Detected confidential")
```

Ta thấy cấu trúc của một luật có dạng như sau:

Rule Header	Rule Option
-------------	-------------

Diễn giải:

Tất cả các Luật của Snort về logic đều gồm 2 phần: Phần header và phần Option.

- Phần Header chứa thông tin về hành động mà luật đó sẽ thực hiện khi phát hiện ra có xâm nhập nằm trong gói tin và nó cũng chứa các tiêu chuẩn để áp dụng luật với gói tin đó.
- Phần Option chứa một thông điệp cảnh báo và các thông tin về các phần của gói tin dùng để tạo nên cảnh báo. Phần Option chứa các tiêu chuẩn phụ thêm để đối sánh luật với gói tin. Một luật có thể phát hiện được một hay nhiều hoạt động thăm dò hay tấn công. Các luật thông minh có khả năng áp dụng cho nhiều dấu hiệu xâm nhập.

Dưới đây là cấu trúc chung của phần Header của một luật Snort:

Action	Protocol	Address	Port	Direction	Address	Port
--------	----------	---------	------	-----------	---------	------

- Action: là phần qui định loại hành động nào được thực thi khi các dấu hiệu của gói tin được nhận dạng chính xác bằng luật đó. Thông thường, các hành động tạo ra một cảnh báo hoặc log thông điệp hoặc kích hoạt một luật khác.

- Protocol: là phần qui định việc áp dụng luật cho các packet chỉ thuộc một giao thức cụ thể nào đó. Ví dụ như IP, TCP, UDP...
- Address: là phần địa chỉ nguồn và địa chỉ đích. Các địa chỉ có thể là một máy đơn, nhiều máy hoặc của một mạng nào đó. Trong hai phần địa chỉ trên thì một sẽ là địa chỉ nguồn, một sẽ là địa chỉ đích và địa chỉ nào thuộc loại nào sẽ do phần Direction "<->" qui định.
- Port: xác định các cổng nguồn và đích của một gói tin mà trên đó luật được áp dụng.
- Direction: phần này sẽ chỉ ra đâu là địa chỉ nguồn, đâu là địa chỉ đích.
- Ví dụ:

alert icmp any any -> any any (msg: "Ping with TTL=100"; ttl: 100;)

Phần đứng trước dấu mở ngoặc là phần Header của luật còn phần còn lại là phần Option. Chi tiết của phần Header như sau:

- Hành động của luật ở đây là "alert": một cảnh báo sẽ được tạo ra nếu như các điều kiện của gói tin là phù hợp với luật (gói tin luôn được log lại mỗi khi cảnh báo được tạo ra).
- Protocol của luật ở đây là ICMP tức là luật chỉ áp dụng cho các gói tin thuộc giao thức ICMP.
- Địa chỉ nguồn và đích của luật ở đây là "any any" tức là luật áp dụng cho tất cả các địa chỉ nguồn và đích.
- Direction của luật ở đây là "->" tức là luật áp dụng cho các gói tin đi từ địa chỉ nguồn đến địa chỉ đích.
- Port của luật ở đây là "any any" tức là luật áp dụng cho tất cả các cổng nguồn và đích.
- Option của luật ở đây là "(msg: "Ping with TTL=100"; ttl: 100;)" tức là luật sẽ tạo ra một cảnh báo với thông điệp "Ping with TTL=100" và giá trị TTL của gói tin là 100.

CHƯƠNG 2. NỘI DUNG THỰC HÀNH

2.1 Chuẩn bị môi trường

- 01 máy tính chạy Linux với RAM tối thiểu 2GB, 10GB đĩa cứng có kết nối mạng (LAN hoặc Internet).
- 01 máy tính chạy Kali Linu
- Bộ phần mềm Snort tải tại <https://www.snort.org/downloads>

2.2 Các bước thực hiện

Bước 1: Chuẩn bị hệ thống

- Đặt tên máy Kali Linux: <Mã SV>-Kali

```
(b22dcat253@kali)-[~]
$ date
Wed Mar 12 08:03:33 AM EDT 2025

(b22dcat253@kali)-[~]
$ echo "Dinh Thi Thanh Tam - B22DCAT253"
Dinh Thi Thanh Tam - B22DCAT253
```

- Đặt tên máy Snort: <Mã SV-Tên SV>

```
b22dcat253@tamdtb22at253:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:0e:2c:08 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 10.10.19.148/24 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe0e:2c08/64 scope link
        valid_lft forever preferred_lft forever
b22dcat253@tamdtb22at253:~$ date
Wed Mar 12 10:43:12 AM UTC 2025
b22dcat253@tamdtb22at253:~$
```

minh chung.txt

File Edit View

Người thực hiện: Dinh Thi Thanh Tam
MSV: B22DCAT253
Ngày thực hiện: 12/3/2025

- Kiểm tra kết nối mạng giữa hai máy: từ máy kali, thực hiện ping đến máy cài đặt snort

```

(b22dcat253@kali)-[~]
$ ping -c 4 10.10.19.148

PING 10.10.19.148 (10.10.19.148) 56(84) bytes of data.
64 bytes from 10.10.19.148: icmp_seq=1 ttl=64 time=6.94 ms
64 bytes from 10.10.19.148: icmp_seq=2 ttl=64 time=2.59 ms
64 bytes from 10.10.19.148: icmp_seq=3 ttl=64 time=5.29 ms
64 bytes from 10.10.19.148: icmp_seq=4 ttl=64 time=2.41 ms

— 10.10.19.148 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.412/4.307/6.944/1.902 ms

(b22dcat253@kali)-[~]
$ date
Wed Mar 12 08:03:33 AM EDT 2025

(b22dcat253@kali)-[~]
$ echo "Dinh Thi Thanh Tam - B22DCAT253"
Dinh Thi Thanh Tam - B22DCAT253

```

➔ Ping thành công

Bước 2: Cài đặt và chạy thử Snort

- Cài đặt Snort trên máy bằng lệnh:

```
sudo apt update
```

```
sudo apt install snort-y
```

- Kiểm tra phiên bản Snort:

```
snort -V
```

```

b22dcat253@tamdtb22at253:~$ snort -V
-*) Snort! <*-
Version 2.9.15.1 GRE (Build 15125)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

b22dcat253@tamdtb22at253:~$ date
Wed Mar 12 10:49:44 AM UTC 2025
b22dcat253@tamdtb22at253:~$ _

```

minh chung.txt

File Edit View

Người thực hiện: Dinh Thi Thanh Tam

MSV: B22DCAT253

Ngày thực hiện: 12/3/2025

Bước 3: Viết và triển khai các quy tắc phát hiện tấn công

- Mở file cấu hình Snort để thêm các quy tắc:

```
sudo nano /etc/snort/rules/local.rules
```

- Thêm các quy tắc:

1. Phát hiện gói tin ping:

alert icmp any any -> \$HOME_NET any (msg:"<Mã SV-Tên SV>-Snort phát hiện có các gói Ping gửi đến."; sid:1000001;)

2. Phát hiện quét cổng 80:

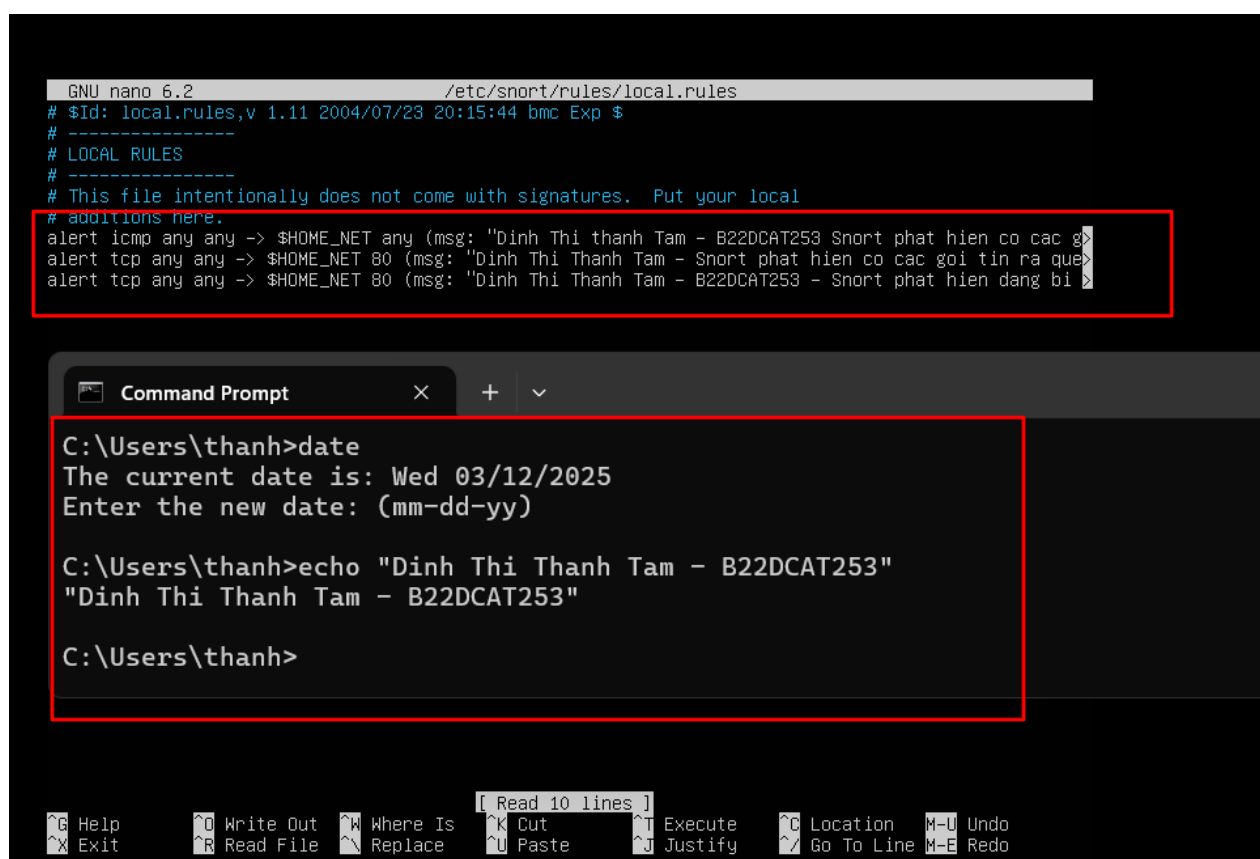
alert tcp any any -> &HOME_NET 80 (msg:"<Mã SV-Tên SV>-Snort phát hiện có các gói tin rà quét trên cổng 80."; sid:1000002;)

3. Phát hiện tấn công TCP SYN Flood:

alert tcp any any -> HOME_NET 80 (flags:S; msg:"<Mã SV-Tên SV>-Snort phát hiện đang bị tấn công TCP SYN Flood."; sid:1000003;)

- Lưu file: ctrl x -> ctrl y, và khởi động lại Snort:

sudo systemctl restart snort



The screenshot shows two windows. The top window is a terminal running GNU nano 6.2, editing the file /etc/snort/rules/local.rules. The file content includes a header with a timestamp and three alert rules. The rules are: 1. Detecting ping packets to the home network. 2. Detecting port scans on port 80. 3. Detecting TCP SYN floods on port 80. The rules use a placeholder &HOME_NET. The bottom window is a Windows Command Prompt. It shows the date command outputting 'Wed 03/12/2025', followed by an echo command outputting 'Dinh Thi Thanh Tam - B22DCAT253'.

```
GNU nano 6.2 /etc/snort/rules/local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert icmp any any -> $HOME_NET any (msg:"Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi tin ra que"
alert tcp any any -> $HOME_NET 80 (msg:"Dinh Thi Thanh Tam - Snort phat hien co cac goi tin ra que"
alert tcp any any -> $HOME_NET 80 (msg:"Dinh Thi Thanh Tam - B22DCAT253 - Snort phat hien dang bi"

C:\Users\thanh>date
The current date is: Wed 03/12/2025
Enter the new date: (mm-dd-yy)

C:\Users\thanh>echo "Dinh Thi Thanh Tam - B22DCAT253"
"Dinh Thi Thanh Tam - B22DCAT253"

C:\Users\thanh>
```

Bước 4: Thực hiện tấn công kiểm thử

1. Kiểm thử phát hiện Ping: Từ máy Kali, sử dụng lệnh ping để ping máy Snort

ping -c 4 <địa chỉ IP máy Snort>

```

(b22dcat253@kali)-[~]
$ ping -c 4 10.10.19.148

PING 10.10.19.148 (10.10.19.148) 56(84) bytes of data.
64 bytes from 10.10.19.148: icmp_seq=1 ttl=64 time=6.94 ms
64 bytes from 10.10.19.148: icmp_seq=2 ttl=64 time=2.59 ms
64 bytes from 10.10.19.148: icmp_seq=3 ttl=64 time=5.29 ms
64 bytes from 10.10.19.148: icmp_seq=4 ttl=64 time=2.41 ms

— 10.10.19.148 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.412/4.307/6.944/1.902 ms

(b22dcat253@kali)-[~]
$ date
Wed Mar 12 08:03:33 AM EDT 2025

(b22dcat253@kali)-[~]
$ echo "Dinh Thi Thanh Tam - B22DCAT253"
Dinh Thi Thanh Tam - B22DCAT253

```

➔ Gửi các gói tin Ping thành công

- Trên máy Snort kiểm tra kết quả phát hiện trên log của Snort:

sudo tail -f /var/log/snort/snort.alert.fast

```

03/12-14:57:37.639952  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.202 -> 10.10.19.148
03/12-14:57:37.640030  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202
03/12-14:57:38.641349  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.202 -> 10.10.19.148
03/12-14:57:38.642141  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202
03/12-14:57:39.643245  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.202 -> 10.10.19.148
03/12-14:57:39.643370  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202
03/12-14:57:40.646220  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.202 -> 10.10.19.148
03/12-14:57:40.646353  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202
03/12-14:57:41.646757  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.202 -> 10.10.19.148
03/12-14:57:41.646884  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202
b22dcat253@tamdtb22at253:~$ date
Wed Mar 12 02:58:04 PM UTC 2025
b22dcat253@tamdtb22at253:~$ echo "Dinh Thi Thanh Tam - B22DCAT253"
Dinh Thi Thanh Tam - B22DCAT253
b22dcat253@tamdtb22at253:~$

```

➔ Cảnh báo Snort:

- Snort phát hiện các gói tin ICMP (ping) gửi đến giữa hai địa chỉ IP (10.10.19.148 ↔ 10.10.19.202).
- Các cảnh báo lặp lại nhiều lần với thông tin về ưu tiên [Priority: 0].

2. Kiểm thử quét cổng 80 bằng nmap:

nmap -sV -p80 -A <địa chỉ IP máy Snort>

```
b22dcat253@kali: ~  
File Actions Edit View Help  
(b22dcat253@kali)-[~]  
$ nmap -sV -p80 -A 10.10.19.148  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-12 10:59 EDT  
Nmap scan report for 10.10.19.148  
Host is up (0.0042s latency).  
  
PORT      STATE SERVICE VERSION  
80/tcp    closed http  
MAC Address: 00:0C:29:0E:2C:08 (VMware)  
Too many fingerprints match this host to give specific OS details  
Network Distance: 1 hop  
  
TRACEROUTE  
HOP RTT      ADDRESS  
1 4.17 ms 10.10.19.148  
  
OS and Service detection performed. Please report any incorrect results at ht  
tps://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 16.60 seconds  
  
(b22dcat253@kali)-[~]  
$ date  
Wed Mar 12 10:59:55 AM EDT 2025  
  
(b22dcat253@kali)-[~]  
$ echo "Dinh Thi Thanh Tam - B22DCAT253"  
Dinh Thi Thanh Tam - B22DCAT253
```

- Quét cổng nmap thành công, trên máy Snort kiểm tra kết quả phát hiện trên log của Snort:

sudo tail -f /var/log/snort/snort.alert.fast

```
03/12-14:59:52.471016  [**] [1:1000003:1] Dinh Thi Thanh Tam - B22DCAT253 - Snort phat hien dang bi  
tan cong TCP SYN Flood! [**] [Priority: 0] {TCP} 10.10.19.202:44402 -> 10.10.19.148:80  
03/12-14:59:52.471016  [**] [1:1000002:1] Dinh Thi Thanh Tam - Snort phat hien co cac goi tin ra que  
t trn cong 80! [**] [Priority: 0] {TCP} 10.10.19.202:44402 -> 10.10.19.148:80  
03/12-14:59:52.495719  [**] [1:1000003:1] Dinh Thi Thanh Tam - B22DCAT253 - Snort phat hien dang bi  
tan cong TCP SYN Flood! [**] [Priority: 0] {TCP} 10.10.19.202:44403 -> 10.10.19.148:80  
03/12-14:59:52.495719  [**] [1:1000002:1] Dinh Thi Thanh Tam - Snort phat hien co cac goi tin ra que  
t trn cong 80! [**] [Priority: 0] {TCP} 10.10.19.202:44403 -> 10.10.19.148:80  
03/12-14:59:52.519444  [**] [1:1000003:1] Dinh Thi Thanh Tam - B22DCAT253 - Snort phat hien dang bi  
tan cong TCP SYN Flood! [**] [Priority: 0] {TCP} 10.10.19.202:44404 -> 10.10.19.148:80  
03/12-14:59:52.519444  [**] [1:1000002:1] Dinh Thi Thanh Tam - Snort phat hien co cac goi tin ra que  
t trn cong 80! [**] [Priority: 0] {TCP} 10.10.19.202:44404 -> 10.10.19.148:80  
03/12-14:59:52.606706  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi  
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202  
b22dcat253@tamdtb22at253:~$ date  
Wed Mar 12 03:01:45 PM UTC 2025  
b22dcat253@tamdtb22at253:~$ echo "Dinh Thi Thanh Tam - B22DCAT253"  
Dinh Thi Thanh Tam - B22DCAT253  
b22dcat253@tamdtb22at253:~$ _
```

➔ Cảnh báo Snort về tấn công SYN Flood:

- Snort phát hiện hệ thống đang bị tấn công TCP SYN Flood.
- Các gói tin tấn công xuất phát từ địa chỉ 10.10.19.202:44402, 44403, 44404 đến 10.10.19.148:80 (cổng HTTP).

- Đây là dấu hiệu của một cuộc tấn công từ chối dịch vụ (DoS) nhằm làm tê liệt dịch vụ web trên cổng 80.

➔ Cảnh báo Snort về gói tin ICMP (ping):

- Các gói tin ICMP (ping) được gửi qua lại giữa 10.10.19.148 và 10.10.19.202.
- Điều này có thể liên quan đến kiểm tra kết nối mạng hoặc tấn công DDoS sử dụng ping flood.

3. Kiểm thử tấn công TCP SYN Flood bằng hping3:

hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source <địa chỉ IP máy Snort>

```
(b22dcat253@kali)-[~]
$ sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 10.10.10.148
[sudo] password for b22dcat253:
Warning: Unable to guess the output interface
HPING 10.10.10.148 (lo 10.10.10.148): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
— 10.10.10.148 hping statistic —
4954482 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

(b22dcat253@kali)-[~]
$ date
Wed Mar 12 11:06:36 AM EDT 2025

(b22dcat253@kali)-[~]
$ echo "Dinh Thi Thanh Tam - B22DCAT253"
Dinh Thi Thanh Tam - B22DCAT253
```

- Tấn công thành công, trên máy Snort kiểm tra kết quả phát hiện trên log của Snort:

sudo tail -f /var/log/snort/snort.alert.fast

```
b22dcat253@tamdtb22at253:~$ sudo tail -f /var/log/snort/snort.alert.fast
03/12-14:59:52.445241  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202
03/12-14:59:52.471016  [**] [1:1000003:1] Dinh Thi Thanh Tam - B22DCAT253 - Snort phat hien dang bi
tan cong TCP SYN Flood! [**] [Priority: 0] {TCP} 10.10.19.202:44402 -> 10.10.19.148:80
03/12-14:59:52.471016  [**] [1:1000002:1] Dinh Thi Thanh Tam - Snort phat hien co cac goi tin ra que
t trn cong 80! [**] [Priority: 0] {TCP} 10.10.19.202:44402 -> 10.10.19.148:80
03/12-14:59:52.495719  [**] [1:1000003:1] Dinh Thi Thanh Tam - B22DCAT253 - Snort phat hien dang bi
tan cong TCP SYN Flood! [**] [Priority: 0] {TCP} 10.10.19.202:44403 -> 10.10.19.148:80
03/12-14:59:52.495719  [**] [1:1000002:1] Dinh Thi Thanh Tam - Snort phat hien co cac goi tin ra que
t trn cong 80! [**] [Priority: 0] {TCP} 10.10.19.202:44403 -> 10.10.19.148:80
03/12-14:59:52.519444  [**] [1:1000003:1] Dinh Thi Thanh Tam - B22DCAT253 - Snort phat hien dang bi
tan cong TCP SYN Flood! [**] [Priority: 0] {TCP} 10.10.19.202:44404 -> 10.10.19.148:80
03/12-14:59:52.519444  [**] [1:1000002:1] Dinh Thi Thanh Tam - Snort phat hien co cac goi tin ra que
t trn cong 80! [**] [Priority: 0] {TCP} 10.10.19.202:44404 -> 10.10.19.148:80
03/12-14:59:52.606706  [**] [1:1000001:1] Dinh Thi thanh Tam - B22DCAT253 Snort phat hien co cac goi
ping gui den! [**] [Priority: 0] {ICMP} 10.10.19.148 -> 10.10.19.202
03/12-15:10:32.124375  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad
Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
03/12-15:10:42.171623  [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad
Traffic] [Priority: 2] {UDP} 0.0.0.0:68 -> 255.255.255.255:67
^C
b22dcat253@tamdtb22at253:~$ date
Sun Mar 23 09:11:37 AM UTC 2025
b22dcat253@tamdtb22at253:~$ echo "Dinh Thi Thanh Tam - B22DCAT253"
Dinh Thi Thanh Tam - B22DCAT253
b22dcat253@tamdtb22at253:~$
```

➔ Nội dung chính log bao gồm:

1. Cảnh báo Snort về gói ICMP (ping)

- Snort phát hiện các gói ICMP (ping) giữa 10.10.19.148 và 10.10.19.202.
- Đây có thể là một cuộc tấn công Ping Flood hoặc kiểm tra kết nối mạng.

2. Cảnh báo Snort về tấn công SYN Flood

- Snort phát hiện cuộc tấn công TCP SYN Flood nhắm vào cổng 80 (HTTP) của địa chỉ 10.10.19.148.
- Các gói SYN đến từ 10.10.19.202:44402, 44403, 44404 → 10.10.19.148:80.
- SYN Flood là một loại tấn công từ chối dịch vụ (DoS) nhằm làm quá tải tài nguyên của máy chủ.

3. Cảnh báo Snort về lưu lượng UDP đáng ngờ

- Snort phát hiện gói tin UDP từ 0.0.0.0:68 đến 255.255.255.67.
- Được phân loại là "Potentially Bad Traffic" (Lưu lượng tiềm ẩn nguy hiểm).

KẾT LUẬN

Bài thực hành cung cấp các kiến thức và kỹ năng cần thiết về hệ thống phát hiện xâm nhập mạng (NIDS), đặc biệt là công cụ Snort. Qua quá trình thực hiện:

- Tìm hiểu tổng quan về hệ thống phát hiện tấn công, các loại IDS và các kỹ thuật phát hiện xâm nhập.
- Cài đặt và cấu hình thành công Snort trên máy chủ Linux.
- Viết và áp dụng các quy tắc Snort để phát hiện các dạng tấn công phổ biến như ping sweep, port scanning và TCP SYN Flood.
- Thực hiện kiểm tra bằng cách giả lập các cuộc tấn công từ máy Kali Linux và quan sát kết quả phát hiện trên giao diện terminal hoặc log của Snort.

Bài thực hành giúp hiểu rõ hơn về cách thức vận hành và triển khai một hệ thống IDS trong thực tế, từ đó nâng cao khả năng giám sát và bảo vệ hệ thống mạng trước các nguy cơ tấn công.

TÀI LIỆU THAM KHẢO

- [1] Chương 5, Giáo trình Cơ sở an toàn thông tin, Học viện Công nghệ BVCT, 2020.
- [2] Suricata: <https://suricata.io/documentation/>
- [3] Snort: <https://www.snort.org/#documents>
- [4] OSSEC: <https://www.ossec.net/docs/>
- [5] Wazuh: <https://documentation.wazuh.com/current/index.html>